

1 Speech Recognition and Signal Analysis by Exact Fast  
2 Search of Subsequences with Maximal Confidence  
Measure

3

4 SPECIFICATION

5 1 TITLE OF THE INVENTION

6 Speech Recognition and Signal Analysis by Exact Fast Search of Subsequences with Maximal  
7 Confidence Measure

8 2 REFERENCE TO APPENDIX SUBMITTED ON CD

9 Not Applicable

10 3 CROSS-REFERENCE TO RELATED APPLICATION

11 This patent application has as parent application the patent application C99-00214/25.02.1999  
12 registered with the State Office for Inventions and Trademarks (OSIM) in Bucharest, Ro-  
13 mania. The present application is the US national stage of the international application  
14 PCT/IB00/00189 registered with the International Patent Office in Geneva.

15 4 BACKGROUND OF THE INVENTION

16 4.1 FIELD OF THE INVENTION

17 The invention relates to a common component of:

- 18 • Speech Recognition, more particularly to the fields of Keyword Spotting and decoding,
- 19 • Segments Alignment for DNA and proteins,
- 20 • Recognition of Objects in Images,

21 4.2 DESCRIPTION OF THE RELATED ART

22 This invention addresses the problem of *keyword spotting (KWS)* in unconstrained speech  
23 without explicit modeling of non-keyword segments (typically done by using filler HMM  
24 models or an ergodic HMM composed of context dependent or independent phone models  
25 without lexical constraints). Several methods (sometimes referred to as “sliding model meth-  
26 ods”) tackling this type of problem have already been proposed in the past. E.g., they use  
27 Dynamic Time Warping (DTW) or Viterbi matching allowing relaxation of the (begin and  
28 endpoint) constraints. These are known to require the use of an “appropriate” normaliza-  
29 tion of the matching scores since segments of different lengths have then to be compared.  
30 However, given this normalization and the relaxation of begin/endpoints, straightforward  
31 Dynamic Programming (DP) is no longer optimal (or, in other words, the DP optimality  
32 principle is no longer valid) and has to be adapted, involving more memory and CPU. In-  
33 deed, at any possible ending time  $e$ , the match score of the best warp and start time  $b$  of  
34 the reference has to be computed (for all possible start times  $b$  associated with unpruned

35 paths). Finally, this adapted DP quickly becomes even more complex (or intractable) for  
36 more advanced scoring criteria (such as the confidence measures mentioned below).

37 Work in the field of confidence level, and in the framework of hybrid HMM/ANN systems  
38 has shown that the use of accumulated local posterior probabilities (as obtained at the  
39 output of a multilayer perceptron) normalized by the length of the word segment (or, better,  
40 involving a double normalization over the number of phones and the number of acoustic  
41 frames in each phone) was yielding good confidence measures and good scores for the re-  
42 estimation of  $N$ -best hypotheses. However, so far the evaluation of such confidence measures  
43 involved the estimation and rescoring of  $N$ -best hypotheses.

44 KWS methods without filler models have in common the selection of a subsequence of  
45 the utterance to match the interesting keyword models. Let  $X = \{x_1, x_2, \dots, x_n, \dots, x_N\}$   
46 denote the sequence of acoustic vectors in which we want to detect a keyword, and let  $M$   
47 be the HMM model of a keyword  $M$  and consisting of  $L$  states  $\mathcal{Q} = \{q_1, q_2, \dots, q_\ell, \dots, q_L\}$ .  
48 Assuming that  $M$  is matched to a subsequence  $X_b^e = \{x_b, \dots, x_e\}$  ( $1 \leq b \leq e \leq N$ ) of  $X$ ,  
49 and that we have an implicit (not modeled) *garbage/filler state*  $q_G$  preceding and following  
50  $M$ , one can define (approximate) the log posterior of a model  $M$  given a subsequence  $X_b^e$  as  
51 the average posterior probability along the optimal path, i.e.:

$$\begin{aligned} 52 \quad -\log P(M|X_b^e) &\simeq \frac{1}{e-b+1} \min_{\forall Q \in M} -\log P(Q|X_b^e) \\ 53 \quad &\simeq \frac{1}{e-b+1} \min_{\forall Q \in M} \{-\log P(q^b|q_G) \\ 54 \quad &\quad - \sum_{n=b}^{e-1} [\log P(q^n|x_n) + \log P(q^{n+1}|q^n)] \\ 55 \quad &\quad - \log P(q^e|x_e) - \log P(q_G|q^e)\} \end{aligned} \tag{1}$$

56 where  $Q = \{q^b, q^{b+1}, \dots, q^e\}$  represents one of the possible paths of length  $(e-b+1)$  in  $M$ , and

57  $q^n$  the HMM state visited at time  $n$  along  $Q$ , with  $q^n \in Q$ . In this expression,  $q_G$  represents  
 58 the “garbage” (filler) state which is simply used here as the non-emitting initial and final  
 59 state of  $M$ . Transition probabilities  $P(q^b|q_G)$  and  $P(q_G|q^e)$  can be interpreted as the keyword  
 60 entrance and exit penalties, but can be simply set to 1. Local posteriors  $P(q_\ell|x_n)$  can be  
 61 estimated using any of the known techniques: multi-gaussians, code-books, or as output  
 62 values of a multilayer perceptron (MLP) used in hybrid HMM/ANN systems. For a specific  
 63 sub-sequence  $X_b^e$ , expression (1) can easily be estimated by dynamic programming since the  
 64 sub-sequence and the associated normalizing factor  $(e - b + 1)$  are given. However, in the  
 65 case of keyword spotting, this expression should be estimated for all possible begin/endpoint  
 66 pairs  $\{b, e\}$  (as well as for all possible word models), and we define the matching score of  $X$   
 67 on  $M$  as:

$$68 \quad S(M|X) = -\log P(M|X_b^{e^*}) \quad (2)$$

69 where the optimal begin/endpoints  $\{b^*, e^*\}$ , and the associated optimal path  $Q^*$ , are the  
 70 ones yielding the lowest average local posterior:

$$71 \quad \langle Q^*, b^*, e^* \rangle = \operatorname{argmin}_{\{Q, b, e\}} \frac{-1}{e - b + 1} \log P(Q|X_b^e) \quad (3)$$

72 Of course, in the case of several keywords, all possible models will have to be evaluated.  
 73 A double averaging involving the number of frames per phone and the number of phones  
 74 usually yields slightly better performance when used to rescore N-best candidates:

$$75 \quad \langle Q^*, b^*, e^* \rangle = \quad (4)$$

$$76 \quad \operatorname{argmin}_{\{Q, b, e\}} \frac{-1}{J} \sum_{j=1}^J \left( \frac{1}{e_j - b_j + 1} \sum_{n=b_j}^{e_j} \log P(q_j^n|x_n) \right) \text{nonumber} \quad (5)$$

77 where  $J$  represents the number of phones in the hypothesized keyword model and  $q_j^n$  the

78 hypothesized phone  $q_j$  for input frame  $x_n$ . However, given the time normalization and  
 79 the relaxation of begin/endpoints, straightforward DP is no longer optimal and has to be  
 80 adapted, usually involving more memory and CPU.

81 Filler-based KWS need a simpler decoding step. Although various solutions have been  
 82 proposed towards the direct optimization of (2), most of the keyword spotting approaches  
 83 today prefer to preserve the optimality and simplicity of Viterbi DP by modeling the complete  
 84 input and explicitly or implicitly modeling non-keyword segments by using so called filler or  
 85 garbage models as additional reference models. In this case, we assume that non-keyword  
 86 segments are modeled by extraneous garbage models/states  $q_G$  (and grammatical constraints  
 87 ruling the possible keyword/non-keyword sequences).

88 Let us consider only the case of detecting one keyword per utterance at a time. In this  
 89 case, the keyword spotting problem amounts at matching the whole sequence  $X$  of length  
 90  $N$  onto an extended HMM model  $\overline{M}$  consisting of the states  $\{q_G, q_1, \dots, q_L, q_G\}$ , in which  
 91 a path (of length  $N$ ) is denoted  $\overline{Q} = \{\overbrace{q_G, \dots, q_G}^{b-1}, q^b, q^{b+1}, \dots, q^e, \overbrace{q_G, \dots, q_G}^{N-e}\}$  with  $(b-1)$  garbage  
 92 states  $q_G$  preceding  $q^b$  and  $(N-e)$  states  $q_G$  following  $q^e$ , and respectively emitting the vector  
 93 sequences  $X_1^{b-1}$  and  $X_{e+1}^N$  associated with the non-keyword segments.

94 Given some estimation of  $P(q_G|x_n)$  (e.g., using probability density functions trained on  
 95 non keyword utterances), the optimal path  $\overline{Q}^*$  (and, consequently  $b^*$  and  $e^*$ ) is then given  
 96 by:

$$\begin{aligned}
 97 \quad \overline{Q}^* &= \underset{\forall \overline{Q} \in \overline{M}}{\operatorname{argmin}} -\log P(\overline{Q}|X) \\
 98 \quad &= \underset{\forall \overline{Q} \in \overline{M}}{\operatorname{argmin}} \{-\log P(Q|X_b^e)} \\
 99 \quad &\quad - \sum_{n=1}^{b-1} \log P(q_G|x_n) - \sum_{n=e+1}^N \log P(q_G|x_n)\} \tag{6}
 \end{aligned}$$

100 which can be solved by straightforward DP (since all paths have the same length). The main  
101 problem of filler-based keyword spotting approaches is then to find ways to best estimate  
102  $P(q_G|x_n)$  in order to minimize the error introduced by the approximations. Sometimes this  
103 value was defined as the average of the  $N$  best local scores while, in other approaches, this  
104 value is generated from explicit filler HMMs. However, these approaches will usually not  
105 lead to the “optimal” solution given by (2).

## 106 5 BRIEF SUMMARY OF THE INVENTION

107 The invention belongs to the technical domain of decoding, classification, alignment and  
108 matching of data.

109 The invention introduces a new method performing tasks in keyword spotting in utter-  
110 ances, detection of subsequences in chains of organic matter (DNA and proteins) and recog-  
111 nition of objects in images. The proposed methods search in an optimized way the matching  
112 that maximizes, over all the possible matchings, certain confidence measures based on nor-  
113 malized posteriors. Three such confidence measures are used, two existed in previous work  
114 in Speech Recognition, and the third one is a new one.

115 Application fields for this invention are: man-machine interfaces (using speech recogni-  
116 tion; ex: control systems, banking, flight services, etc), coordination systems (for industrial  
117 robots and automata) and development systems for pharmaceutic products.

118 6 BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE  
119 DRAWINGS

120 Not Applicable

121 7 DETAILED DESCRIPTION OF THE INVENTION

122 In the following, we show that it is possible to define an iterative process, referred to  
123 as *Iterating Viterbi Decoding (IVD)* with good/fast convergence properties, estimating the  
124 value of  $P(q_G|x_n)$  such that straightforward DP (6) yields exactly the same segmentation  
125 (and recognition results) than (3). While the same result could be achieved through a  
126 modified DP in which all possible combinations (all possible begin/endpoints) would be  
127 taken into account, the method proposed below is much more efficient (in terms of both  
128 CPU and memory requirements).

129 Compared to previously devised “sliding model” methods the first method proposed here  
130 is based on:

- 131 1. A matching score defined as the average observation probability (posterior) along the  
132 most likely state sequence. It is indeed believed that local posteriors are more appro-  
133 priate to the task.
- 134 2. The iteration of a Viterbi decoding algorithm, which does not require scoring for all  
135 begin/endpoints or N-best rescoring, and which can be proved to (quickly) converge to  
136 the “optimal” (from the point of view of the chosen scoring functions) solution without

137       requiring any specific filler models, using straightforward Viterbi alignments (similar  
138       to regular filler-based KWS, but for some versions at the cost of a few iterations).

139       The IVD method is based on a similar criterion as the filler based approaches (6), but  
140       rather than looking for explicit (and empirical) estimates of  $P(q_G|x_n)$  we aim at mathe-  
141       matically estimating its value (which will be different and adapted to each utterance) such  
142       that solving (6) is equivalent to solving (3). Thus, we perform an iterative estimation of  
143        $P(q_G|x_n)$ , such that the segmentation resulting of (6) is the same than what would be ob-  
144       tained from (3). Defining  $\varepsilon_t = -\log P(q_G|x_n)$  at iteration  $t$ , the proposed method can be  
145       summarized as follows:

146       1. Start the first iteration,  $t = 0$ , from an initial value  $\varepsilon_0 = \Pi$  (it is actually proven that  
147       the iterative process presented here will always converge to the same solution, in more  
148       or less cycles with the worst case upper bound of  $N$  iterations, independently of this  
149       initialization, e.g., with  $\Pi$  equal with a cheap estimation of the score of a “match”).

150       In one of the developed versions,  $\varepsilon_0$  is initialized to  $-\log$  of the maximum of the local  
151       probabilities  $P(q_k|x_n)$  for each frame  $x_n$ .

152       An alternative choice is to initialize  $\varepsilon_0$  to a pre-defined threshold score,  $T$ , that expres-  
153       sion (1) should reach to declare a keyword “matching” (see step 4 below). In this last  
154       case, if  $\varepsilon_1 > \varepsilon_0$  at the first iteration, then we can (as proven) directly infer that the  
155       match will be rejected, otherwise it will be accepted.

156       2. Given the estimate  $\varepsilon_t$  of  $P(q_G|x_n)$  at current iteration  $t$ , find the optimal path  $\langle \overline{Q}_t, b_t, e_t \rangle$   
157       according to (6) and matching the complete input.

158     3. Estimate the value of  $\varepsilon_{t+1}$  to be used in the next iteration as the average of the local  
159     posteriors along the optimal path  $Q_t$  (matching the  $X_{b_t}^{e_t}$  resulting of (6) on the keyword  
160     model) i.e.:

161     
$$\varepsilon_{t+1} = -\frac{1}{(e_t - b_t + 1)} \log P(Q_t | X_{b_t}^{e_t}) \quad (7)$$

162     4. Increment  $t$  and return to (2) iterating until convergence is detected. If we are not  
163     interested in the optimal segmentation, this process could also be stopped as soon as it  
164     reaches a  $\varepsilon_{t+1}$  lower than a (pre-defined) minimum threshold,  $T$ , below which we can  
165     declare that a keyword has been detected.

166     Correctness and convergence proof of this process and generalization to other criteria, are  
167     available: each IVD iteration (from the second iteration) will decrease the value of  $\varepsilon_t$ , and the  
168     final path yields the same solution than (3). The above method has a very good experimental  
169     convergence speed (3-5 iterations in our tests). For one version of IVD (when  $\varepsilon_0$  is initialized  
170     using the acceptance threshold,  $T$ ), the detection is decided after one single step.

171     A version with the same effort but suboptimal results is proposed in the following para-  
172     graph. Let  $T(\bar{M}, X)$  be a matrix holding the HMM emission probabilities for an utterance  
173      $X$  whose time-frames define the columns, and where the states of the hypothesized word  
174      $W$  define the rows. When using the standard DP, one computes for each element of the  
175     matrix  $T(\bar{M}, X)$  at frame  $k$  of  $X$  and state  $s$  of  $\bar{M}$  three values:  $S_{ks}$ ,  $L_{ks}$  and  $C_{ks}$ , where  
176      $S_{ks}$  corresponds to the sum of the entries on the optimal path that leads to the entry,  $L_{ks}$   
177     holds the length of the optimal path computed so far, and  $C_{ks}$  is the estimation of the cost  
178     on the optimal expanded path. By a path leading to an entry  $T(k, s)$  we mean a sequence  
179     of entries in the table  $T$ , such that there is exactly an entry for each time frame  $t \leq k$ . At

180 each entry  $T(k, s)$ , DP selects a locally optimal path noted  $P_{ks}$ . At each step  $k$ , we consider  
 181 all pairs of entries of table  $T(\bar{M}, X)$  of type  $T(k, s)$ ,  $T(k - 1, t)$ . We update for each such  
 182 pair, the current cost  $C_{ks}$  (initially  $\infty$ ), by comparing it with the alternative given by:

$$S_{ks} = S_{(k-1)t} - \log p(s|x_k)p(s|t)$$

183

$$L_{ks} = L_{(k-1)t} + 1, \forall t > 0, t \leq L$$

184

$$C_{ks} = \frac{S_{ks}}{L_{ks}} \quad (8)$$

186 wanting to have at step  $k$  the path  $P_{ks}$  from the paths  $P_{(k-1)t}$  that minimizes  $C_{NL}$ . With  
 187 DP, one will choose the  $P_{ks}$  with minimal  $C_{ks}$ .

188 This version can yield suboptimal results since the optimality principle is not respected  
 189 by the expression 8. The optimality principle of Dynamic Programming requires that the  
 190 path to the frame  $k - 1$  that minimizes  $C_{NL}$ , also minimizes  $C_{ks}$  for an entry at frame  $k$  of  
 191 table  $T(\bar{M}, X)$ .

192 Another technique that is suboptimal in time and/or quality is obtained from the previous  
 193 one adopting a beam-search approach and a set of safe prunings. The Dynamic Programming  
 194 can be viewed as a set of safe prunings that are applied at each entry of the DP table and  
 195 has the property that only one alternative is maintained. Dynamic Programming cannot be  
 196 used, since the principle of optimality is not respected. The following types of safe pruning  
 197 that can be done are introduced by the present invention. Within the current invention we  
 198 found a set of safe prunings as follows: we have proved that if at a frame  $a$  we have two paths  
 199  $P'_a$  and  $P''_a$  with  $S''_a < S'_a$  and  $L'_a < L''_a$ , then at no frame  $c \geq a$  will a path  $P''_c$  be forsaken for  
 200 a path  $P'_c$  if  $P'_a \subset P'_c$ ,  $P''_a \subset P''_c$  and  $P'_c \setminus P'_a \equiv P''_c \setminus P''_a$ . We will note the order relation as  $P''_a \prec P'_a$ .

201 We have further shown that a path  $P'$  may be safely discarded only when we know a lower  
202 cost one,  $P''$ .

203 
$$P' \prec P'' \Rightarrow C'_k < C''_k \quad (9)$$

204 Thus, the method described in following method computes  $S(M, X)$  and  $Q^*$  from equa-  
205 tion (3). By ordering the set of paths, according to Equation 9, we only need to check the  
206 step (1.1) of the following method up to the eventual insertion place. The last paths are  
207 candidates for pruning in step (1.2). In order for the pruning to be acceptable, we will prune  
208 only paths that were too long on the last state. An additional counter for each path is  
209 needed for storing the state length. This counter is reset when an entry from another row  
210 is added and is incremented at each advance with a frame. The following steps detail this  
211 method for a model  $W$  and an utterance  $X$ :

212 a) Initialize all elements of a matrix,  $\text{SetOfPaths}(1..N, 1..K)$ , to  $\emptyset$

213 b) For all frames from 1 to  $N$ , for all states from 1 to  $K$ , for all candidates  $p_i$  in  
214  $\text{SetOfPaths}(\text{frame-1}, 1..K)$ :

215 – For all  $p_j$  in  $\text{SetOfPaths}[\text{frame}, \text{state}]$ , if  $p_i \prec p_j$  then delete  $p_j$  (1.1), and if  $p_j \prec p_i$   
216 then continue step b) (1.2)

217 – Insert  $p_i$  in  $\text{SetOfPaths}[\text{frame}, \text{state}]$

218 c) Select  $\text{SetOfPaths}[\text{frame}, K]$  as the best of the candidates

219 The next method builds on the previous technique and is a fast procedure for maximizing  
220 a more complex confidence measure that yields better results in practice. The corresponding

221 confidence measure is defined as:

$$222 \quad \frac{1}{NVP} \sum_{h_i \in VP} \frac{\sum_{pst \in h_i} - \log(pst)}{length(h_i)} \quad (10)$$

223 where NVP stands for the *number of visited phonemes* and VP stands for the *set of visited*  
224 *phonemes*. An average is computed over all posteriors  $pst$  of the emission probabilities for the  
225 time frames matched to the visited phoneme  $h_i$ . The function  $length(h_i)$  gives the number of  
226 time frames matched against  $h_i$ . This method uses a breath first Beam Search algorithm. It  
227 exploits a set of reduction rules and certain normalizations. For the state  $q_G$ , in this method,  
228 the logarithm of the emission posterior is equal with zero. For each frame  $e$  and for each  
229 state  $s$ , the set of paths/probabilities of having the frame  $e$  in the state  $s$  is computed as  
230 the first  $\mathcal{N}$  maxima ( $\mathcal{N}$  can be finite) of the confidence measure for all paths in HMM  $\overline{M}$  of  
231 length  $e$  and ending in the state  $s$ . The paths that according to the reduction rules will loose  
232 the final race when compared with another already known path, will be deleted as well. Let  
233 us note  $a_1, p_1, l_1$ , respectively  $a_2, p_2$  and  $l_2$  the confidence measure for the previously visited  
234 phonemes, the posterior in the current phoneme and the length in the current phoneme for  
235 the path  $Q_1$ , respectively the path  $Q_2$ . The rules that can be used for the reduction of the  
236 search space by discarding a path  $Q_1$  for a path  $Q_2$  are in this case any of the next ones:

237     1.  $l_2 \geq l_1, A > 0, B \leq 0$  and  $L_c^2 A + L_c B + C \geq 0$

238     2.  $l_2 \geq l_1, A \geq 0, B \geq 0$  and  $C \geq 0$

239     3.  $l_2 \geq l_1, A \leq 0, C \geq 0$  and  $L^2 A + LB + C \geq 0$

240     4.  $l_2 \geq l_1, A = 0, B < 0$  and  $LB + C \geq 0$

241 where  $A = a_1 - a_2$ ,  $B = (a_1 - a_2)(l_1 + l_2) + p_1 - p_2$ ,  $C = (a_1 - a_2)l_1l_2 + p_1l_2 - p_2l_1$ ,  $L =$   
242  $L_{max} - \max\{l_1, l_2\}$ ,  $L_c = -B/2A \geq 0$  and  $L_{max}$  is the maximum acceptable length for a  
243 phoneme. By discarding paths only if one of the above rules is satisfied, the optimum defined  
244 by the confidence measure with double normalization can be guaranteed, if no phone may be  
245 avoided by the HMM  $M$ . Any HMM may be decomposed in HMMs with this quality. The  
246 4-th rule is included in the 3-rd and its test is useless if the last one was already checked.  
247 The first test,  $l_2 \geq l_1$  tells us if  $Q_2$  has chances to eliminate  $Q_1$ , otherwise we will check  
248 if  $Q_1$  eliminates  $Q_2$ . These tests were inferred from the conditions of maintaining the final  
249 maximal confidence measure while reduction takes place. In order to use the method of  
250 double normalization without decomposing HMMs that skip some phonemes, the previous  
251 rules are modified taking into account the number of visited phonemes for any path  $F_1$   
252 respectively  $F_2$  and the number of phonemes that may follow the current state. A simplified  
253 test can be:

254 •  $l_2 \geq l_1$ ,  $A \geq 0$ ,  $p_1 \geq p_2$  respectively  $F_2 \geq F_1$  for the HMMs that skips phonemes.

255 This test is weaker than the 2<sup>nd</sup> reduction rule. For example a path is eliminated by a second  
256 path if the first one has an inferior confidence measure (higher in value) for the the previous  
257 phonemes, a shorter length and the minus of the logarithm of the cumulated posterior in  
258 the current phoneme also inferior (higher in value) to that of the second one. An additional  
259 confidence measure based on the maximal length,  $L_{max}$ , and on the maximum of the minus  
260 of the logarithm of the cumulated and normalized posterior in phoneme,  $P_{max}$ , can be used  
261 in order to limit the number of stored paths.

262 •  $p > L_{max}P_{max}$  in any state

- 263     •  $\frac{p}{l} > P_{max}$  at the output from a phoneme
- 264 where p and l are the values in the current phoneme for the minus of the logarithm of  
 265 cumulated posterior and for the length of the path that is discarded. These tests allow for  
 266 the elimination of the paths that are too long without being outstanding, respectively of  
 267 the paths with phonemes having unacceptable scores, otherwise compensated by very good  
 268 scores in other phonemes. If  $N$  is chosen equal with one, the aforementioned rules are no  
 269 longer needed, but always we propagate the path with the maximal current estimation of  
 270 the confidence measure. The obtained results are very good, even if the defined optimum is  
 271 guaranteed for this method only when  $N$  is bigger than the length of the sequence allowed  
 272 by  $L_{max}$  or of the tested sequence. The same approach is valid for the simple normalization,  
 273 where the HMM for the searched word will be grouped into a single phoneme.
- 274     The present invention can exploit a newly designed a confidence measure, version named  
 275 "Real Fitting", that represents differently the exigencies of the recognition. Since the  
 276 phonemes and the absent states can be modeled by the used HMMs, we find it interest-  
 277 ing to request the fitting of each phoneme in the model with a section of the sequence.  
 278 Therefore, we measure the confidence level of a subsequence as being equal with the max-  
 279 imum over all phonemes of the minus of the logarithm of the cumulated posterior of the  
 280 phone, normalized with its length:
- $$281 \quad \max_{phonem \in Visited\ Phonems} \frac{\sum_{phonem} - \log(posterior)}{phonem\ length} \quad (11)$$
- 282 The rule that may be used in this framework for the reduction of the number of visited paths  
 283 is:
- 284     •  $Q_2$  is discarded in favor of another path  $Q_1$  if the confidence measure of the Real

285 Fitting for the previous phonemes is inferior (higher in value) for  $Q_2$  compared with  
286  $Q_1$ , and if  $p_1 \leq p_2$  and  $l_2 \leq l_1$ .

287 where  $p_1, l_1$ , respectively  $p_2, l_2$  represent the minus of the logarithm of the cumulated poste-  
288 rior respectively the number of frames in the current phoneme for the path  $Q_1$  respectively  
289  $Q_2$ . Similarly to the previous method, the set of visited paths can be pruned by discarding  
290 those where:

291 •  $p > L_{max} P_{max}$  in any state

292 •  $\frac{p}{l} > P_{max}$  at the output from a phoneme

293 where  $p$  and  $l$  are the values in the current phoneme for the minus of the logarithm of the  
294 cumulated posterior and for the length of the path that is discarded. We recall that the  
295 meaning of the constants are the maximal length  $L_{max}$ , respectively the accepted maxima  
296 of the minus of the logarithm of the cumulated and normalized posterior in phoneme,  $P_{max}$ .

297 This invention thus proposes a new method for keyword spotting, based on recent ad-  
298 vances in confidence measures, using local posterior probabilities, but without requiring the  
299 explicit use of filler models. A new method, referred to as *Iterating Viterbi Decoding (IVD)*,  
300 to solve the above optimization problem with a simple DP process (not requiring to store  
301 pointers and scores for all possible ending and start times). Other three new beam-search  
302 algorithms corresponding to three different confidence measures are also proposed.

303 To summarize, the object of the invention consists of:

304 • Method of recognition of a subsequence using a direct maximization of confidence  
305 measures.

306       • The method of IVD for directly maximizing the confidence measures based on simple  
307           normalization.

308       • The use of the confidence measure and method of recognition named ‘Real Fitting’,  
309           based on individual fitting for each phoneme.

310       • Methods of recognition using simple and double normalization by:

311       • combining these measures with additional confidence measures mentioned here, respec-  
312           tively the maximal length and real matching limitation.

313       • The use of the aforementioned methods in keyword recognition.

314       • The use of the aforementioned methods in subsequence recognition of organic matter.

315       • The use of the aforementioned methods in recognition of objects in images.

#### 316       DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

317       Execution: The method can be performed using a personal computer or can be imple-  
318           mented in specialized hardware.

319       1. A representation under the form of an HMM is obtained for the subsequences that are  
320           looked for (word, protein profile, section of an image of the object).

321       2. A tool will be obtained (eventually trained Ex: for speech recognition) for the esti-  
322           mation of the posteriors. For example multi-Gaussians, neuronal networks, clusters,  
323           database with Generalized Profiles and mutation matrices (PAM, BLOSSUM, etc.).

324     3. One of the proposed algorithms should be implemented. They yield close performance  
325       but the method of Real Fitting coupled with a well checked dictionary should perform  
326       best.

327     For the first algorithm (IVD)

328     (a) The classic algorithm of Viterbi is implemented with the modification that, for  
329       each pair  $P = \langle sample, state \rangle$  one propagates the time-frame of transition be-  
330       tween the state  $q_G$  and the states of the HMM  $M$  for the path that arrives at P.  
331       These are inherited from the path that wins the entrance in the pair P, excepting  
332       for the moment when their decision is taken, namely when they receive the index  
333       of the corresponding sample.

334     (b)  $w = -\log P(M|X_b^e)$  is computed by subtracting from the cumulated posterior  
335       that is returned by the Viterbi algorithm for the path  $Q_{b_t}^{e_t}$ , the value  $(N - (e_t -$   
336        $b_t + 1)) * \varepsilon_t$  corresponding to the contribution of the states  $q_G$  and dividing the  
337       result through  $e_t - b_t + 1$ .  $e_t - b_t + 1$  from the previous formula can be factored  
338       outside the fraction.

339     (c) The initialization of  $\varepsilon$  is made with an expected mean value. One can use the  $w$   
340       that is computed when the state  $q_G$  is associated with an emission posterior equal  
341       to the average of the best  $K$  emission probabilities of the current sample as done  
342       in the well-known “garbage on-line model”. In this case, K is trained using the  
343       corresponding technique.

344     The next ‘Beam search’ algorithms, are implemented according to the description in

345 the corresponding sections. For each pair  $P = \langle sample, state \rangle$  one computes for each  
346 corresponding path the sum and length in the last phoneme, as well as the sum over  
347 the normalized cumulated posteriors of the previous phonemes (and their number).

348 Also, the entrance and exit samples into the HMM  $M$  are computed and propagated  
349 like in the previous method, in order to ensure the localization of the subsequence.

350 4. If one searched entity (keyword, sequence, object) can have several HMM models, all  
351 of them are taken into consideration as competitors. This is the case of the words  
352 with several pronunciations (or of the objects that have different structures in different  
353 states, for the recognition in images).

354 After the computation of the confidence measure for each model of the subsequences,  
355 one eliminates those with a confidence measure in disagreement with a 'threshold' that  
356 is trained for the configuration and the goal of the given application. For example, for  
357 speech recognition with neuronal networks and minus of the logarithm of the posteriors,  
358 the 'threshold' is chosen in the wanted point of the ROC curve obtained in tests.

359 5. The remained alternatives are extracted in the order of their confidence measure and  
360 with the elimination of the conflicting alternatives until exhaustion. Each time when  
361 an alternative is eliminated, the searched entity with the corresponding HMM is re-  
362 estimated for the remaining sections in the sequence in which the search is performed.  
363 If the new confidence measure passes the test of the 'threshold', then it will be inserted  
364 in the position corresponding to its score in the queue of alternatives.

365 6. The successful alternatives can undergo tests of superior levels like for example a

366 question of confirmation for speech recognition, opinion of one operator, etc.

367 7. For objects recognition in images:

368 Posteriors are obtained by computing a distance between the color of the model and  
369 that of element in the section of the image. If the context requires, the image will be  
370 preprocessed to ensure a certain normalization (Ex: changeable conditions of light will  
371 make necessary a transformation based on the histogram).

372 The phonemes of the speech recognition correspond to parts of the object. The struc-  
373 ture (existence of transitions and their probabilities) can be modified, function of the  
374 characteristics detected along the current path. For example, after detecting regions  
375 of the object with certain lengths, one can estimate the expected length of the remain-  
376 ing regions. Thus, the number of the expected samples for the future states can be  
377 established and the HMM attached to the object will be configured accordingly.

378 A direction is scanned for the detection of the best fitting and afterwards, other direc-  
379 tions will be scanned for discovering new fittings, as well as for testing the previous  
380 ones. The final test will be certified by classical methods such as cross-correlation or  
381 by the analysis of the contours in the hypothesized position.

382 To mention some examples for the application of the proposed method:

383 • The recognition of keywords begins to be used in answering automates of banking  
384 system as well as telephone and automates for control, sales or information. The  
385 method offers a possibility to recognize keywords in spontaneous speech with multiple  
386 speakers.

- 387     • The recognition of DNA sequences is important for the study of the human Genome.
- 388         One of the biggest problem of the involved techniques consists in the high quantity of
- 389         data that have to be processed.
- 390     • The recognition of objects in images is used, among others, in cartography and in the
- 391         coordination of industrial robots. The method allows a quick estimation of the position
- 392         of the objects in scenes and can be validated with extra tests, using classical methods
- 393         of cross-correlation.